

## My Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Point Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	x . . . . .	5
3.1.2.2	y . . . . .	5
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	geometry.cpp File Reference . . . . .	7
4.1.1	Function Documentation . . . . .	7
4.1.1.1	intersection() . . . . .	7
4.1.1.2	orientationTriangleArea() . . . . .	7
4.1.1.3	savePair() . . . . .	8
4.2	geometry.h File Reference . . . . .	8
4.2.1	Function Documentation . . . . .	9
4.2.1.1	intersection() . . . . .	9
4.2.1.2	orientationTriangleArea() . . . . .	9
4.2.1.3	savePair() . . . . .	9
4.2.2	Variable Documentation . . . . .	10

4.2.2.1	click_counter	10
4.2.2.2	intersectionPoints	10
4.2.2.3	p1	10
4.2.2.4	p2	10
4.2.2.5	points	10
4.3	main.cpp File Reference	10
4.3.1	Detailed Description	11
4.3.2	Function Documentation	11
4.3.2.1	changeSize()	11
4.3.2.2	main()	11
4.3.2.3	myMouse()	12
4.3.2.4	renderScene()	12
4.3.3	Variable Documentation	12
4.3.3.1	click_counter	12
4.3.3.2	height	12
4.3.3.3	intersectionPoints	13
4.3.3.4	p1	13
4.3.3.5	p2	13
4.3.3.6	points	13
4.3.3.7	width	13
	<b>Index</b>	<b>15</b>

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Point</a> . . . . .	<a href="#">5</a>
---------------------------------	-------------------



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">geometry.cpp</a>	7
<a href="#">geometry.h</a>	8
<a href="#">main.cpp</a>	10





## Chapter 3

# Class Documentation

### 3.1 Point Struct Reference

```
#include <geometry.h>
```

#### Public Attributes

- float [x](#)
- float [y](#)

#### 3.1.1 Detailed Description

Struct to facilitate manipulation of x and y coordinates.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 [x](#)

```
float Point::x
```

##### 3.1.2.2 [y](#)

```
float Point::y
```

The documentation for this struct was generated from the following file:

- [geometry.h](#)



## Chapter 4

# File Documentation

### 4.1 geometry.cpp File Reference

```
#include "geometry.h"
```

#### Functions

- float [orientationTriangleArea](#) ([Point](#) a, [Point](#) b, [Point](#) c)
- int [intersection](#) (void)
- void [savePair](#) ([Point](#) p1, [Point](#) p2)

#### 4.1.1 Function Documentation

##### 4.1.1.1 intersection()

```
void intersection (  
    void )
```

Computes if there is an intersection between all pairs of segments.

If the intersection exists, its coordinates are calculated. It uses the orientation method to determine whether the points are on different sides

##### 4.1.1.2 orientationTriangleArea()

```
float orientationTriangleArea (  
    Point a,  
    Point b,  
    Point c )
```

Computes twice the signed area of the orientation triangle. If it is oriented counterclockwise, the area is positive. Otherwise, it is negative.

**Parameters**

<i>a</i>	a <a href="#">Point</a> struct
<i>b</i>	a <a href="#">Point</a> struct
<i>c</i>	a <a href="#">Point</a> struct

**Returns**

Two times the area of the orientation triangle

**4.1.1.3 savePair()**

```
void savePair (
    Point p1,
    Point p2 )
```

Saves a pair of points (starting point and end point) of a segment.

It saves the pair of points in a vector.

**Parameters**

<i>p1</i>	a <a href="#">Point</a> struct
<i>p2</i>	a <a href="#">Point</a> struct

## 4.2 geometry.h File Reference

```
#include <vector>
#include <iostream>
#include <GL/glut.h>
```

**Classes**

- struct [Point](#)

**Functions**

- float [orientationTriangleArea](#) ([Point](#) a, [Point](#) b, [Point](#) c)
- int [intersection](#) (void)
- void [savePair](#) ([Point](#) p1, [Point](#) p2)

## Variables

- [Point](#) p1
- [Point](#) p2
- `vector< pair< Point, Point > > points`
- `vector< Point > intersectionPoints`
- `int click_counter`

## 4.2.1 Function Documentation

### 4.2.1.1 intersection()

```
int intersection (
    void )
```

Computes if there is an intersection between all pairs of segments.

If the intersection exists, its coordinates are calculated. It uses the orientation method to determine whether the points are on different sides

### 4.2.1.2 orientationTriangleArea()

```
float orientationTriangleArea (
    Point a,
    Point b,
    Point c )
```

Computes twice the signed area of the orientation triangle. If it is oriented counterclockwise, the area is positive. Otherwise, it is negative.

#### Parameters

<i>a</i>	a <a href="#">Point</a> struct
<i>b</i>	a <a href="#">Point</a> struct
<i>c</i>	a <a href="#">Point</a> struct

#### Returns

Two times the area of the orientation triangle

### 4.2.1.3 savePair()

```
void savePair (
    Point p1,
    Point p2 )
```

Saves a pair of points (starting point and end point) of a segment.

It saves the pair of points in a vector.

#### Parameters

<i>p1</i>	a <a href="#">Point</a> struct
<i>p2</i>	a <a href="#">Point</a> struct

## 4.2.2 Variable Documentation

### 4.2.2.1 click\_counter

```
int click_counter
```

### 4.2.2.2 intersectionPoints

```
vector<Point> intersectionPoints
```

### 4.2.2.3 p1

```
Point p1
```

### 4.2.2.4 p2

```
Point p2
```

### 4.2.2.5 points

```
vector<pair<Point,Point> > points
```

## 4.3 main.cpp File Reference

```
#include "geometry.h"
```

## Functions

- void `changeSize` (int w, int h)
- void `myMouse` (int button, int state, int x, int y)
- void `renderScene` (void)
- int `main` (int argc, char \*\*argv)

## Variables

- float `width` = 640
- float `height` = 480
- int `click_counter` = 0
- `Point` p1
- `Point` p2
- vector< pair< `Point`, `Point` > > `points`
- vector< `Point` > `intersectionPoints`

### 4.3.1 Detailed Description

Trabalho 1 de Computação Gráfica I

#### Author

Lucas Rodrigues Teixeira Nunes

### 4.3.2 Function Documentation

#### 4.3.2.1 `changeSize()`

```
void changeSize (  
    int w,  
    int h )
```

Resizes the viewport when the window size changes.

#### Parameters

<i>w</i>	an integer variable
<i>h</i>	an integer variable

#### 4.3.2.2 `main()`

```
int main (
```

```
int argc,  
char ** argv )
```

#### 4.3.2.3 myMouse()

```
void myMouse (   
    int button,  
    int state,  
    int x,  
    int y )
```

Callback function that gets mouse coordinates and button state.

##### Parameters

<i>button</i>	an integer variable
<i>state</i>	an integer variable
<i>x</i>	an integer variable
<i>y</i>	an integer variable

#### 4.3.2.4 renderScene()

```
void renderScene (   
    void )
```

Callback function to render OpenGL scene and draw the segments and intersections.

### 4.3.3 Variable Documentation

#### 4.3.3.1 click\_counter

```
int click_counter = 0
```

#### 4.3.3.2 height

```
float height = 480
```



#### 4.3.3.3 intersectionPoints

```
vector<Point> intersectionPoints
```

#### 4.3.3.4 p1

```
Point p1
```

#### 4.3.3.5 p2

```
Point p2
```

#### 4.3.3.6 points

```
vector<pair<Point,Point> > points
```

#### 4.3.3.7 width

```
float width = 640
```



# Index

- changeSize
  - main.cpp, [11](#)
- click\_counter
  - geometry.h, [10](#)
  - main.cpp, [12](#)
- geometry.cpp, [7](#)
  - intersection, [7](#)
  - orientationTriangleArea, [7](#)
  - savePair, [8](#)
- geometry.h, [8](#)
  - click\_counter, [10](#)
  - intersection, [9](#)
  - intersectionPoints, [10](#)
  - orientationTriangleArea, [9](#)
  - p1, [10](#)
  - p2, [10](#)
  - points, [10](#)
  - savePair, [9](#)
- height
  - main.cpp, [12](#)
- intersection
  - geometry.cpp, [7](#)
  - geometry.h, [9](#)
- intersectionPoints
  - geometry.h, [10](#)
  - main.cpp, [12](#)
- main
  - main.cpp, [11](#)
- main.cpp, [10](#)
  - changeSize, [11](#)
  - click\_counter, [12](#)
  - height, [12](#)
  - intersectionPoints, [12](#)
  - main, [11](#)
  - myMouse, [12](#)
  - p1, [13](#)
  - p2, [13](#)
  - points, [13](#)
  - renderScene, [12](#)
  - width, [13](#)
- myMouse
  - main.cpp, [12](#)
- orientationTriangleArea
  - geometry.cpp, [7](#)
  - geometry.h, [9](#)
- p1
  - geometry.h, [10](#)
  - main.cpp, [13](#)
- p2
  - geometry.h, [10](#)
  - main.cpp, [13](#)
- Point, [5](#)
  - x, [5](#)
  - y, [5](#)
- points
  - geometry.h, [10](#)
  - main.cpp, [13](#)
- renderScene
  - main.cpp, [12](#)
- savePair
  - geometry.cpp, [8](#)
  - geometry.h, [9](#)
- width
  - main.cpp, [13](#)
- x
  - Point, [5](#)
- y
  - Point, [5](#)